What is claimed is:

1. A system for normalizing information content, the system comprising:

a template normalizer for matching and applying a template to the information content; and

an automatic normalizer for folderizing the information content;

wherein the template normalizer attempts to match a template to the information content, and if not, the automatic normalizer folderizes the information content to produce a normalized information content.

2. The system of claim 1 wherein the template normalizer recognizes patterns in the information content, and wherein the template normalizer dynamically changes the information content to match the template.

3. The system of claim 2 wherein the template normalizer determines if the variation in the information content is too great to match to a template, and if so, forwards the information content to the automatic normalizer.

4. The system of claim 1 wherein the template normalizer utilizes regular expression pattern matching to the information content.

5. The system of claim 1 wherein the information content is represented by a document object tree, and wherein the template normalizer utilizes regular expression pattern matching to the document object tree.

6. The system of claim 1 wherein the automatic normalizer utilizes normalization markup embedded in the information content to provide the automatic normalizer with specific instructions.

7. The system of claim 1 wherein the automatic normalizer utilizes meta-tags embedded in the information content to provide the automatic normalizer with at least one specific instruction.

8. The system of claim 7 wherein the specific instruction is to create a folder containing a portion of the information content.

9. The system of claim 7 wherein the specific instruction is to trigger markup based normalization.

10. The system of claim 1 wherein the template normalizer utilizes normalization markup embedded in the information content to provide the template normalizer with at least one specific instruction.

11. The system of claim 10 wherein the specific instruction is to trigger markup based normalization.

12. The system of claim 1 wherein the normalization markup does not affect the page for display by a PC-based browser that utilizes hypertext markup language (HTML).

13. The system of claim 1 wherein the information content is in the form of a document object model (DOM).

14. The system of claim 1 wherein the information content is in the form of a document object tree.

15. The system of claim 1 further comprising:

a QDOM for generating a document object tree, wherein the document object tree is represented by a mutable object.

16. A system for normalizing a document tree representation, the system comprising:

an automatic normalizer for applying pattern recognition and weighting heuristics on the document tree to produce a normalized document tree.

17. The system of claim 16 wherein the automatic normalizer comprises a markup assisted normalizer for processing normalization markup in the document tree to produce the normalized document tree.

18. The system of claim 17 wherein the normalization markup does not affect a page for display by a PC-based browser that utilizes hypertext markup language (HTML).

19. The system of claim 16 wherein the normalized document tree represents a hierarchical representation of information in the document tree.

20. A system for normalizing a document tree representation, the system comprising:

a template normalizer for matching a document tree to a template tree and applying changes to the document tree to produce a normalized document tree.

21. The system of claim 20 wherein the template normalizer determines if the document tree matches the template tree, and if not, forwards the document tree to an automatic normalizer.

22. The system of claim 20 wherein the template normalizer utilizes regular expression pattern matching to match the template tree to a document tree.

23. A method for normalizing information content, the method comprises:

matching and applying a template to the information content, and if unsuccessful:

determining if the information content contains normalization markup, and if so:

utilizing normalization markup in the information content to normalize the information content.

24. The method of claim 23 further comprises:

determining if the information content contains normalization markup, and if not:

applying pattern recognition and weighting heuristics on the information content to normalize the information content.

25. The method of claim 23 further comprises:

recognizing patterns in the information content, wherein the template normalizer dynamically changes the information content to match the template.

26. The method of claim 23 further comprises:

determining if a variation in the information content is too great to match the template, and if so:

determining if the information content contains normalization markup.

27. A method for automatically normalizing a document tree, the method comprises:

determining node weighting criteria;

weighting nodes in the document tree according to the determined criteria; and

determining parent-child relationships between the weighted nodes based on the weighted nodes to produce a normalized document tree.

28. The method of 27 further comprises:

weighting nodes in a table; and

attempting to match the table to a predefined pattern of weights, and if successful:

extracting data in response to the predefined pattern.

29. The method of 28 further comprises:

attempting to match the table to a predefined pattern of weights, and if unsuccessful:

extracting data according to the weighted nodes.

30. The method of claim 27 further comprises:

applying changes to the document tree according to a normalization markup comprising dropping nodes, moving nodes, partitioning nodes into folders, and calling user defined formatting rules on the nodes.

31. A method for generating a document object tree, the method comprises:

receiving data; and

storing information relating to the data into a plurality of arrays;

wherein the plurality of arrays utilize re-usable buffers, and wherein the stored information describes the document object tree and tree dependencies as a mutable object.

32. The method of claim 31 further comprises:

transforming the document object tree, wherein the transformed document object tree is represent by the single mutable object.

33. The method of claim 31 further comprises:

adding an array to the plurality of arrays as the received data grows in size.

34. The method of claim 31 wherein the plurality of arrays are used to hold values that represent properties of a node of the document object tree.

35. The method of claim 31 further comprises:

referencing a re-usable content buffer that contains data;

wherein the plurality of arrays store start and end positions of data that reference the data stored in the re-usable content buffer.

36. The method of claim 31 wherein the plurality of arrays contain values associated with nodes of the data, and wherein operations on the nodes can be carried out by utilizing the value as referenced to the affected nodes.

37. The method of claim 31 further comprises:

normalizing the document object tree model by a template normalizer for applying templates to the document object tree.

38. The method of claim 31 further comprises:

normalizing the document object tree model by an automatic normalizer for applying pattern recognition and weighting heuristics on the document tree to produce a normalized document tree..

McDonnell Boehnen Hulbert & Berghoff
300 S. Wacker Drive, Floor 3200
Chicago, IL 60606
Telephone: 312-913-0001

39. A method for comparing a document tree against a template tree, the method further comprises:

matching the document tree by utilizing a template markup language comprising regular expression; and

applying changes to the document tree according to the template markup language.

40. The method of claim 39 wherein the document tree is represented by a plurality of nodes, and wherein the template markup language comprises dropping at least one of the plurality of nodes, moving at least one of the plurality of nodes, partitioning at least one of the plurality of nodes into folders, and calling user defined formatting rules on at least one of the plurality of nodes.